#README - Online Auction Site

Group 16

Snehal Ajakkala (sa1544)
Devin Ling (dtl65)
Ethan Nguyen (evn11)
Ruan Jude Angelo Jamalod (rtj36)

Implementation

This project is a makeshift implementation of an online auction site. For our implementation, we use HTML for the user interface, MySQL for the database server, and Java, Javascript, and JDBC for connectivity between the user interface and database server.

For our entities, we initially create several tables:

`users`

The list of users. It ha sthe type of user that someone using the site could be. This includes end-users (buyers and sellers) and staff (admins and customer representatives). The primary key is **email**, since every user must have a unique email address. They also have a username.

NOTE: in our case, username is more similar to a password in real-world scenarios. We did not include a `password` field since `username` was sufficient for our implementation.

`auctions`

The list of auctions. Each auction has an **auctionID**, which is the primary key. It holds information, such as when the auction closes and the bidding increments. It also includes a reference to the item it's auctioning via, **itemID**, the primary key of `items`.

`items`

The list of items. Each item has an **itemID**, which is the primary key. It holds information, such as the type of item, its size, and its color.

`bids`

The list of bids. Each bid had a **bidID**, which is the primary key. It holds information, such as the amount of the bid, the owner of the bid, and whether it is an auto-bid or not. It also includes a reference to the auction that the bid is for via, **auctionID**, as well as a reference to the owner of the bid, via **email** of `users`.

`customerServ`

The list of questions (and answers). End-users ask questions and customer representatives can answer them. Each question and answer is one instance of this table. They each have a **questionID**. The table holds information, such as what the question is, the answer, the user who asked the question, and the representative who answered the question. Thus, it has references to **email** of `users` for both the asking user and the customer rep.

`itemsReq`

The list of requested items. End-users can request specific items, and all other end-users can see such items. Each request has a **requestID**. The table holds information about the characteristics of the requested item, such as the color, size, and type of item. It has a reference to the end-user who put in the request, and thus their **email**.

Usage

In order to use the site, one must create an account with an email and username. Any end-user can do so when immediately brought to the first page.

After logging in, a user has several options listed for them in the form of buttons. All of the actions they can perform will bring them to various pages on the website, where they can view or fill out more information.

For staff, a previous administrator account is created. The credentials are as follows:

Email:	admin@gmail.com
Username:	admin

Thanks for reading (and grading) our project! Hopefully it meets expectations, as we had some fun creating it!